AD-A224 223

Copy 18 of 37 copies

IDA DOCUMENT D-590

Ada VALIDATION TESTING TOOLS EVALUATION

Audrey A. Hook
Robert J. Knapper

April 1989

Prepared for
Ada Joint Program Office

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

IDA Log No. HQ 89-034306

DEF ITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract MDA 903 84 C 0031 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

This Document is published in order to make available the material it contains for the use and convenience of interested parties. The material has not necessarily been completely evaluated and analyzed, nor subjected to formal IDA review.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1989	Final

4. TITLE AND SUBTITLE
Ada Validation Testing Tools Evaluation

5. FUNDING NUMBERS
MDA 903 84 C 0031
T-D5-304

6. AUTHOR(S)
Robert J. Knapper

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses (IDA)
1801 N. Beauregard Street
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Document D-590

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Ada Joint Program Office (AJPO)
Room 3D139, The Pentagon
Washington, D.C. 20301

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, unlimited distribution.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)

IDA Document D-590, Ada Validation Testing Tools Evaluation, presents the results of an Institute for Defense Analyses (IDA) evaluation of the Ada Compiler Validation Capability (ACVC) Tailoring Tool and the Computer-Assisted Ada Validation Tool (CAV). This document will be used as a guide to initiate enhancements to the validation tools that were evaluated (ACVC and CAV). It is directed towards the Ada Validation Facility (AVF) and the Ada Joint Program Office (AJPO) staff members.

14. SUBJECT TERMS
Ada Programming Language; Ada Validation; Ada Compilers; Ada Validation Tools; Ada Validation Facilities (AVFs); ACVC; CAV.

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

IDA DOCUMENT D-590

Ada VALIDATION TESTING TOOLS EVALUATION

Audrey A. Hook
Robert J. Knapper

April 1989

INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 84 C 0031
Task T-D5-304

TABLE OF CONTENTS

PREFACE AND ACKNOWLEDGEMENTS

The purpose of IDA Document D-590, *Ada Validation Testing Tools Evaluation,* is to present the results of IDA's evaluation of the Ada Compiler Validation Capability Tailoring Tool and the Computer-assisted Ada Validation tool.

The importance of this document is based on IDA's continuing technical support under Task Order T-D5-304, Ada Validation, for the Ada Joint Program Office's efforts in maintaining and improving the Ada validation process. D-590 will be used as a guide to initiate enhancements to the validation tools that were evaluated. As a document, D-590 is directed towards Ada Validation Facility personnel and AJPO staff members.

The document was reviewed on 27 March 1989 by the members of the following CSED Peer Review: Mr. Terry Mayfield, Dr. Reginald Meeson, and Ms. Katydean Price.

The document was externally reviewed by Dr. William Dashiell - NIST, Mr. Bobby Evans - Wright-Patterson AFB, Mr. Adrian Groom - NCC, Dr. Stephan Heilbrunner - IABG, and Dr. Jacqueline Sidi - AFNOR.

Ada Validation Testing Tools Evaluation

1. INTRODUCTION

The Ada Joint Program Office is seeking to maintain the uniform and efficient operation of its Ada Validation Facilities (AVFs) as an ever increasing number of validations are performed each year. Two tools, the Ada Compiler Validation Capability (ACVC) Tailoring Tool and the Computer-assisted Ada Validation Tool (CAV), have been developed by the United Kingdom and French validation facilities respectively, as a contribution toward improvements in AVF operations.

The Tailoring Tool, developed by the National Computing Centre (NCC) in the United Kingdom, "customizes" the ACVC test suite for a particular implementation and generates an information file about the customized suite. CAV, developed by the Association Française de Normalisation (AFNOR) in France, uses this information file and, in conjunction with the computer-readable results from validation testing, provides an automatic capability for analyzing a validation.

In Section 3, the paper provides a brief background into the validation process and where automated tools may be of assistance. Section 4 describes the purpose and function of the validation tools. Section 5 presents the evaluation results. Lastly, Section 6 discusses recommendations for changes or enhancements to the tools.

2. SCOPE

IDA conducted an evaluation of the Tailoring Tool and CAV to determine the fitness of these tools to help improve AVF operations. Over a four month period, IDA assumed the role of both the implementor and an AVF by developing and running a validation testing scenario. This scenario allowed the validation tools to be exercised in near-actual conditions. The recommendations presented in the last section of this paper are based on the results of running the scenario.

3. BACKGROUND

The ACVC test suite (version 1.10) contains over 4,000 files that make up over 3,700 tests. During the lifetime of an ACVC version, tests may be found to contain errors or it may be determined that one or more tests violate the Ada Programming Language Standard, ANSI/MIL-STD-1815A. In either case, the test is withdrawn from the suite and is no longer required to be passed for validation. In addition, some tests are designed to check capacities, limits, and other implementation-dependent features. These tests may be ruled inapplicable to some implementations. Also, some of the tests require specific implementation defined values to be checked. Finally, some tests contain multiple test objectives and require that the test be split into two or more subtests on some implementations to demonstrate that the objectives are met. Thus, each implementation validates using a customized version of the test suite to account for the variances listed above. For the most part, the customization of this suite by both the implementor and the AVF has been with little or no automation. The NCC Tailoring Tool provides this needed automation.

The validation process has two major steps, pre-validation testing and on-site formal validation testing. The pre-validation step involves the testing of the implementation by the implementor using the customized test suite. The test results that are generated are forwarded to the AVF for analysis. To ensure a thorough analysis, each result is examined and checked against the expected result.

If the pre-validation results are approved, the on-site testing should be *pro forma*. In the past, the on-site results were as thoroughly checked by hand as were the pre-validation results. This should not be necessary. Instead of hand checking the on-site results, a more efficient method would be to compare the on-site results with the approved pre-validation results. Such a process essentially involves character string matching and file comparison, both of which can be automated. The AFNOR CAV tool provides this automation.

4. VALIDATION TESTING TOOLS

4.1 Tailoring Tool

The Tailoring Tool has two distinct functions, suite customizing and suite information generation. For both of these functions, the tool requires the user to generate the following input files:

a. FNAMES - The entire list of *filenames* in the ACVC test suite, in Ada Language Reference Manual chapter order, alphabetized within each test class.

b. MACROS - The list of test *macros* and their associated values.

c. SPLITS - The list of tests that are *split* and the filenames associated with each individual split.

d. WDRNS - The list of filenames that are *withdrawn* from the test suite.

e. SPCLS - The list of filenames that require *special* examination.

f. INAPPS - The list of filenames that have been ruled *inapplicable* for the implementation.

g. TSINFO - The list of implementation specific *test information* containing the number of applicable, inapplicable, and withdrawn tests for each class/chapter pair.

In addition, the tool requires that the entire test suite reside in one directory along with the tool itself.

When the tool begins its execution, the user is queried for information. The user is first asked for the date and the name of the implementation being validated. The user is then asked whether or not the implementation supports filenames longer than eight characters and whether or not the implementation supports three-character filename extensions. After answering these questions, the tool builds internal tables from the supplied input files, then proceeds to perform macro substitutions into all of the ACVC tests requiring it. The tool generates these customized tests based on the specific macro values provided by the user in the file MACROS.

The tool opens and closes every file listed in FNAMES to make sure that all of the files exist. Since no directory qualifiers are allowed in FNAMES, all of the files must reside in the same directory as the tool. If files are missing, the tool alerts the user, both on the screen and in an error file. Once the tool establishes that all of the files exist, a completeness count, based on the information in the file TSINFO, is performed. Any discrepancies are noted on the screen and in

the error file.

Finally, the tool generates TTINF, an information file about the test suite. For each test, TTINF contains a test name, lists all the subfilenames if any, indicates if a test contains macro substitutions, is inapplicable, etc. This file is used as part of the input to CAV. In addition, a report tracing the activities of the tool and its results is generated. Any errors that were encountered during the tool's execution are included in this report.

4.2 Computer-assisted Ada Validation Tool

CAV provides an AVF with the capability to examine, analyze, and compare validation results with automated assistance. The tool requires the following input files:

a. Test result file description - The character patterns used to compare against the actual test results.

b. Test suite description file (TTINF) - The information file created by the Tailoring Tool.

c. User interface files - The set of files used to create the command menus.

d. Terminal control sequences file - The file containing control characters for screen manipulation.

e. Initialization file - The file containing the names of all of the other files used by CAV.

In addition, the compilation, link, or execution results, as appropriate for each test in the ACVC suite, will be needed in computer readable form. CAV requires that each individual test result be contained within a separate file.

With the exception of the result files provided by the implementor and the TTINF file generated by the Tailoring Tool, the other files are included with the tool in a generic form. They can be modified for use on a particular system for a particular implementation. For example, the test result file description file contains the pattern PASSED, to be searched for at the end of an executable result. This is a standard pattern since it is generated by the ACVC report package. Link results, or compilation results have patterns that differ between implementations. These patterns would have to be entered into the test result file description file.

CAV starts its execution by reading the initialization file and setting the control sequences as specified in the control sequences file. The validation management menu is then displayed. From this menu, pre-validations and on-site validation results can be set up into database files maintained by CAV. These files merge together the information from the TTINF file and the result files. The menu also allows examination of validation information, such as who's validation it is, what test is currently being analyzed, etc. Finally, the menu can take the user to additional menus for operating on the test results.

The test operations menu provides the selections for analyzing and comparing test results. The results can be accessed individually or as one or more sets. The purpose of analyzing a test result is to set the test's status. This status can be any one of the following:

a. WITHDRAWN - A test on the withdrawn list receives this status.

b. ANOMALOUS - A test that exhibits an unusual result where conformity is indeterminant receives this status.

c. INAPPLICABLE - A test that has been ruled inapplicable to the implementation receives this status.

d. FAILED - A test that does not produce the expected result(s) receives this status.

e. PASSED - A test that produces the expected result(s) receives this status.

For executable test results, this status is nominally the actual result printed by the ACVC report package. For compilation or link results, this status is determined by examining the results.

Comparing test results involves a direct file comparison between the pre-validation and on-site validation results. If the results match, the test status is entered as PASSED, regardless of what the result of the two files were. For example, if both a pre-validation result and an on-site result are FAILED, then a comparison will yield PASSED to indicate that the result is acceptable. If the results do not match, the test status is entered as FAILED to indicate that the result is not acceptable. An analysis or comparison report is written to a log file that is kept for each validation. This report contains the status for each test.

5. EVALUATION

The tools were evaluated to determine how easy they were to install and make operational, how friendly the user interface was, how complete and clear the documentation was, etc. Four computer systems under three different Ada runtime systems (Sun 3/280, Sun 3/180 with VERDIX Ada Development System (VADS) 5.5, Sun 3/50 with VADS 5.41, and VAX 8600 with DEC VAX Ada 1.5) were used in the evaluation. All of the systems were used to compile the tools, providing a portability measure. Because of time and other resource constraints, only the Sun 3/280 was used for checking the execution of the tools.

A random sample of approximately 1,000 ACVC class C tests was used to create a validation scenario. Class C tests were used because they produce predictable results that would facilitate the evaluation of the tools. The tests included some that had been withdrawn and some that would be inapplicable to the VADS compiler used in the evaluation. Validation testing was then performed and the results were collected.

5.1 Tailoring Tool

The Tailoring Tool was evaluated first for two reasons. The first reason is that this tool is much smaller and simpler than CAV. The second reason is more obvious. The TTINF file that the tool generates is needed by CAV. The tool was executed twice. The first time was with the entire ACVC suite, the second was with the random sample of 1,000 C tests.

5.1.1 Installation

a. File Transfer - The tool source files on the IBM compatible floppy disk were spooled off through an IBM-PC/AT to IDA's Sun server system using the *ftp* utility. No problems were noted in the transfer. However, the tool should be made available on magnetic tape for installation onto systems without floppy disk drive access.

b. Compilation - All of the Tailoring Tool source files compiled and linked on the Sun systems and the VAX 8600. The files on the floppy included a compilation order and identified the name of the unit to link. The following statistics were collected:

(1) Compilation times:

Sun 3/280 VADS 5.5	-	2.5 min
Sun 3/180 VADS 5.5	-	6.0 min
Sun 3/50 VADS 5.41	-	7.0 min
VAX 8600 DEC VAX Ada 1.5	-	3.0 min

(2) Errors or warnings noted - None.

(3) Sizes

(a) Source size - 80.7Kb

(b) Executable size - 204.8Kb (on the Sun systems)

(4) Portability - The Tailoring Tool was ported, without changes, to all four of the systems used.

5.1.2 Use

a. Execution - The tool started operating without any problems. The user interface queried for some information, built its internal tables, then proceeded to open up files and make macro subsitutions.

b. Input error detection/retry capability - The tool recovered if incorrect input was given while querying the user for input at the start of execution. This was, however, without any indication of error. The user was simply re-prompted for information. The tool did not recover if one of the required input files was missing, aborting on an unhandled exception. The tool then had to be restarted from the beginning.

c. Correct functioning - The tool did not function correctly when first installed. The reason for this was a lack of testing its functionality on more than one system. The tool was developed on an MS-DOS based system that can only handle 8-character filenames. Although the tool was supposed to handle more than 8-character filenames, it did not. In addition, the filenames that were generated for the macro substituted files had two blanks at the end of the extension. Some systems would treat those blanks as significant characters which was not intended. The code was modified to correct these problems. The tool then functioned as expected.

d. User friendliness - The tool has a limited functionality and the user interface is reflective of this. Very little interaction is required of the user.

5.1.3 Documentation Quality

The hard copy documentation for the tool was clear and concise. No deviations appeared from what was written and what actually occurred during execution except as noted in section 5.1.2.c.

5.2 Computer-assisted Ada Validation Tool

5.2.1 Installation

a. File Transfer - The tool source files on the IBM compatible floppy disk were spooled off through an IBM-PC/AT to IDA's Sun server system using the *ftp* utility. No problems were noted in the transfer. However, the tool should be made available on magnetic tape for

installation onto systems without floppy disk drive access.

b. Compilation - All of the CAV source files compiled and linked correctly with one exception. A bug in the VADS compiler on the Sun systems that mistakingly required an initial value for a record discriminant, made it necessary to modify one file. Since the error was with the compiler and not the CAV source, this was not viewed negatively with respect to CAV. The files on the floppy included a compilation order and identified the name of the unit to link. The following statistics were collected:

   (1) Compilation times:

Sun 3/280 VADS 5.5	-	12.5 min
Sun 3/180 VADS 5.5	-	29.0 min
Sun 3/50 VADS 5.41	-	unknown, the system ran out of disk
VAX 8600 DEC VAX Ada 1.5	-	31.5 min

   (2) Errors or warnings noted - One file failed to compile due to a compiler error on the Sun systems. A workaround to the problem was found by providing an initial value for a particular record discriminant. The VADS compiler also issued several warnings for extraneous loop index declarations. These warnings did not affect the compilability of the code.

   (3) Sizes

      (a) Source size - 649.7Kb

      (b) Executable size - 466.9Kb (on the Sun systems)

   (4) Portability - CAV did not port to the Sun 3/50. A VADS compiler error that caused an infinite loop and exhaustion of free disk space is suspected. The other Sun systems used a newer version of VADS and compiled the tool without error. No errors occurred in porting the tool to the VAX 8600.

5.2.2 Use

a. Execution - The menu options invoked the appropriate functions. The only questionable choice of action occurred during compare operations. If both the pre-validation result status and the on-site result status were FAILED, the status was changed to PASSED when the results were compared. Although the compare was done correctly, the status change might be interpreted incorrectly.

b. Input error detection/retry capability - The tool did not abort as a result of any incorrect input and the user was always prompted to try again.

c. User friendliness - The menu-driven tool provided a good interface to the user. On-line assistance might have been useful, but as selections were made, the user was always informed of what options were available.

5.2.3 Documentation Quality

The hard copy documentation was adequate. However it was difficult to understand with a first reading, especially the initial steps necessary to make the tool operational. Each menu option was explained, but no complete examples were given to show the actual functionality. This led to

a trial and error usage of the tool the first time. After some experience with the tool, the documentation was easier to follow.

6. CONCLUSIONS AND RECOMMENDATIONS

Both the Tailoring Tool and CAV performed as specified in their documentation. No major deficiencies were found. However, the Tailoring Tool had to be modified as documented in section 5.1.2.c. The tools would help to reduce the amount of time needed to perform an on-site validation.

While the overall evaluation of the tools is positive, some implementation requirements were noted for change. These changes would improve the tools' effectiveness.

a. Tailoring Tool

(1) Multiple directory allowance - Currently, the tool requires that the ACVC test suite reside in a single directory with the tool itself. This requirement could prove to be bothersome on some systems because of disk space or directory table size limitations. At the very minimum, the tool should allow for distributing the test suite into subdirectories based on the test classes to improve the tool's portability. This change would be a minor code modification, involving less than 200 lines of code.

(2) Eliminate hard-coded filenames - The tool uses several input files whose names have been hard-coded. The use of an initialization file with the names and directory locations of these input files would make the tool more portable. This change would also be a minor modification involving less than 100 lines of additional code.

b. Computer-assisted Ada Validation Tool

(1) Documentation clarification - The documentation for CAV lacked a solid visible example of using the tool. If a complete example of setting up a pre-validation and on-site validation along with using the analyzer and comparator were included, the documentation would be much easier to understand. Excerpts from a real validation in which the tool was utilized could be contained in an appendix for the documentation. This would require a minimum effort.

(2) Comparator test status mark change - If both the pre-validation and on-site validation results for a test are FAILED, then a comparison of these two results yielded a test status of PASSED. A less confusing mark for matched compared results would be MATCHED. This is very minor code modification requiring a minimum of effort and no additional code.

(3) Simplify tool - CAV is very complex. The analyzer part provided a way to examine machine-readable test results one at a time. Although this is a convenient feature, it is not intended as a substitute for hand checking pre-validation test results. The comparator part is the most useful part for moving toward expediting on-site validation. Assuming that the pre-validation results have been checked by hand and their results approved, then using the comparator to check differences between these results and the on-site results would save time. A simpler and smaller tool which was a comparator for the full set of tests would port easier and would give AVFs assistance where it is needed most. No additional code would need to be added, but

stripping out certain parts of the tool would be time consuming and would involve a thorough re-testing of the tool. Unless the comparator already exists as a stand-alone tool, at least one manmonth of effort would be needed.

Alternatively, with a minimum effort, the CAV menus could be changed to reflect a limited functionality while not actually removing the code corresponding to functions not to be used. The tool remains the same size, but with a limited number of options making the tool easier to use. While there is no indication that the tool will not port with its current size, this menu-limiting option is essentially cosmetic and would be of little value if ported to a system that could not handle the size of the tool.

Distribution List for IDA Document D-590

NAME AND ADDRESS	NUMBER OF COPIES
Sponsor	
Dr. John P. Solomond Director Ada Joint Program Office Room 3E114 The Pentagon Washington, D.C. 20301-3081	2
Other	
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
IIT Research Institute 4600 Forbes Blvd., Suite 300 Lanham, MD 20706 Attn. Ann Eustice	1 (Camera-Ready Copy)
Mr. Karl H. Shingler Department of the Air Force Software Engineering Institute Joint Program Office (ESD) Carnegie Mellon University Pittsburgh, PA 15213-3890	1
Dr. Jacqueline Sidi AFNOR Tour Europe, Cedex 7 F-92080, Paris la Defence France	1
Dr. Stephan Heilbrunner IABG, Dept. SZT Einsteinstrasse D-8012 OttoBrunn Federal Republic of Germany	1

NAME AND ADDRESS	NUMBER OF COPIES
Mr. Bobby Evans Ada Validation Facility Standard Languages and Environments Div. Engineering Applications Directorate DCS/Communications-Computer Systems Wright-Patterson AFB Ohio, 45433-6503	1
Mr. Steve Wilson Ada Validation Facility Standard Languages and Environments Div. Engineering Applications Directorate DCS/Communications-Computer Systems Wright-Patterson AFB Ohio, 45433-6503	1
Dr. William Dashiell Ada Compiler Validation Manager Software Standards Validation Group National Institute of Standards and Technology Gaithersburg, MD 20899	1
Ms. Jane Pink Ada Validation Facility Manager National Centre for Information Technology National Computing Centre Ltd. Oxford Road, Manchester M1 7ED England	1
CSED Review Panel	
Dr. Dan Alpert, Director Program in Science, Technology & Society University of Illinois Room 201 912-1/2 West Illinois Street Urbana, Illinois 61801	1
Dr. Thomas C. Brandt 10302 Bluet Terrace Upper Marlboro, MD 20772	1

NAME AND ADDRESS	NUMBER OF COPIES
Dr. Ruth Davis The Pymatuning Group, Inc. 2000 N. 15th Street, Suite 707 Arlington, VA 22201	1
Dr. C.E. Hutchinson, Dean Thayer School of Engineering Dartmouth College Hanover, NH 03755	1
Mr. A.J. Jordano Manager, Systems & Software Engineering Headquarters Federal Systems Division 6600 Rockledge Dr. Bethesda, MD 20817	1
Dr. Ernest W. Kent Philips Laboratories 345 Scarborogh Road Briarcliff Manor, NY 10510	1
Dr. John M. Palms, President Georgia State University University Plaza Atlanta, GA 30303	1
Mr. Keith Uncapher University of Southern California Olin Hall 330A University Park Los Angeles, CA 90089-1454	1
IDA	
General W.Y. Smith, HQ	1
Mr. Philip L. Major, HQ	1
Dr. Robert E. Roberts, HQ	1
Ms. Anne Douville, CSED	1
Ms. Audrey A. Hook, CSED	2
Mr. Robert J. Knapper, CSED	2
Mr. R. Danford Lehman, CSED	1
Mr. Terry Mayfield, CSED	1
Dr. Reginald N. Meeson, CSED	1
Ms. Katydean Price, CSED	2
Dr. Richard L. Wexelblat, CSED	1

NAME AND ADDRESS	NUMBER OF COPIES
IDA Control & Distribution Vault	3